Discordant Voting Processes on Finite Graphs*†

Colin Cooper[1], Martin Dyer[2], Alan Frieze[3], and Nicolás Rivera[4]

1 **Department of Informatics, King's College London, London, UK**
colin.cooper@kcl.ac.uk
2 **School of Computing,University of Leeds, Leeds, UK**
M.E.Dyer@leeds.ac.uk
3 **Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA, USA**
alan@random.math.cmu.edu
4 **Department of Informatics, King's College London, London, UK**
nicolas.rivera@kcl.ac.uk

Abstract

We consider an asynchronous voting process on graphs which we call discordant voting, and which can be described as follows. Initially each vertex holds one of two opinions, red or blue say. Neighbouring vertices with different opinions interact pairwise. After an interaction both vertices have the same colour. The quantity of interest is T, the time to reach consensus, i.e. the number of interactions needed for all vertices have the same colour.

An edge whose endpoint colours differ (i.e. one vertex is coloured red and the other one blue) is said to be discordant. A vertex is discordant if its is incident with a discordant edge. In discordant voting, all interactions are based on discordant edges. Because the voting process is asynchronous there are several ways to update the colours of the interacting vertices.

- *Push:* Pick a random discordant vertex and push its colour to a random discordant neighbour.
- *Pull:* Pick a random discordant vertex and pull the colour of a random discordant neighbour.
- *Oblivious:* Pick a random endpoint of a random discordant edge and push the colour to the other end point.

We show that $\mathbf{E}T$, the expected time to reach consensus, depends strongly on the underlying graph and the update rule. For connected graphs on n vertices, and an initial half red, half blue colouring the following hold. For oblivious voting, $\mathbf{E}T = n^2/4$ independent of the underlying graph. For the complete graph K_n, the push protocol has $\mathbf{E}T = \Theta(n \log n)$, whereas the pull protocol has $\mathbf{E}T = \Theta(2^n)$. For the cycle C_n all three protocols have $\mathbf{E}T = \Theta(n^2)$. For the star graph however, the pull protocol has $\mathbf{E}T = O(n^2)$, whereas the push protocol is slower with $\mathbf{E}T = \Theta(n^2 \log n)$.

The wide variation in $\mathbf{E}T$ for the pull protocol is to be contrasted with the well known model of synchronous pull voting, for which $\mathbf{E}T = O(n)$ on many classes of expanders.

1998 ACM Subject Classification C.2.4 Distributed Systems, F.2 Analysis of algorithms, G.2 Discrete mathematics

Keywords and phrases Distributed consensus, Voter model, Interacting particles, Randomized algorithm

Digital Object Identifier 10.4230/LIPIcs.ICALP.2016.145

* A full version of this paper is available at http://arxiv.org/abs/1604.06884.
† This work was supported in part by EPSRC grant EP/M005038/1, "Randomized algorithms for computer networks", NSF grant DMS0753472,and Becas CHILE.

43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016).
Editors: Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi;
Article No. 145; pp. 145:1–145:13


Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

We consider a type of asynchronous distributed voting process on graphs which we call discordant voting, and which can be described as follows. Initially each vertex holds one of two opinions, red or blue say. Neighbouring vertices of different colours, i.e. whose opinions differ, interact pairwise. After an interaction both vertices have the same colour. If, at some step, all vertices have the same colour, we say that a consensus has been reached.

The problem of reaching consensus in graph by means of local interactions is an abstraction of the behavior of both human society and computer networks. As a consequence the process of voting on graphs has been widely studied. Distributed voting finds application in various fields of computing including consensus and leader election in large networks [4, 12], serialisation of read and write in replicated data-bases [10], and the analysis of social behavior in game theory [21]. Voting algorithms are usually simple, fault-tolerant, and easy to implement [12, 14]. Recently, there has been considerable interest in *population protocols*. In this model the interacting vertices can make limited computations using a finite state machine to address a wide range of problems in distributed computing, see e.g. [2].

The classical model, synchronous pull voting, is reasonably well understood. If the colours of the vertices are initially distinct, the randomized process takes $\Theta(n)$ expected steps to reach consensus on many classes of expander graphs on n vertices. This holds for the complete graph K_n (Aldous [1]), and almost all r-regular random graphs [7]. For general results based on the eigenvalue gap and variance of the degree sequence see [6]. Hassin and Peleg [12] and Nakata *et al.* [17] considered the two-party pull voting model on connected graphs, and discussed its application to consensus problems in distributed systems.

In contrast to the case of synchronous voting, where only the pull protocol is well defined, for asynchronous voting, there are at least three ways to update the colours of the interacting vertices.

- *Push:* Pick a random vertex and push its colour to a random neighbour.
- *Pull:* Pick a random vertex and pull the colour of a random neighbour.
- *Oblivious:* Pick a random endpoint of a random edge and push the colour to the other end point.

Discordant voting originated in the complex networks community as a model of social evolution (see e.g. [11], [18]). The general version of the model allows *rewiring*. The interacting vertices can break edges joining them and reconnect elsewhere. This serves as a model of social interaction in which vertices will either change their opinion or their friends.

Holme and Newman [13] investigated discordant voting as a model of a self-organizing network which restructures based on the acceptance or rejection of differing opinions among social groups. At each step, a random discordant edge uv is selected, and an endpoint $x \in \{u, v\}$ chosen with probability $1/2$. With probability $1 - \alpha$ the opinion of x is pushed to the other endpoint y, and with probability α, y breaks the edge and rewires to a random vertex with the same opinion as itself. Simulations suggested the existence of threshold behavior in α. This was investigated further by Durrett et al. [8] for sparse random graphs of constant average degree 4. The paper studies two rewiring strategies, rewire-to-random, and rewire-to-same, and finds experimental evidence of a phase transition in both cases. Basu and Sly [3] made a formal analysis of rewiring for Erdos-Renyi graphs $G(n, 1/2)$ with $1 - \alpha = \beta/n$, $\beta > 0$ constant. They found that for either strategy, if β is sufficiently small the network quickly disconnects maintaining the initial proportions. As β increases the minority proportion decreases, and in rewire-to-random a positive fraction of both opinions survive.

Although discordant voting seems a natural model of local interaction, its behavior, is

not well understood even in the simplest cases. The aim of this paper is a fundamental study of expected time to consensus in the absence of rewiring. As discordant voting always chooses an edge between the red and blue sets, it should be more efficient, and thus finish faster than an asynchronous pull voting process which ignores this information, and takes $\Omega(n^2)$ steps on many classes of sparse graphs (see [5]). However, we find the performance of discordant voting protocols vary considerably with the structure of the underlying graph, and sometimes in a quite counter-intuitive way.

We suppose that the initial vertex colours in the two-party voting model are red and blue, and let $R(t), B(t)$ denote the sets of vertices with the given colours at any step t. For the oblivious protocol, the expected time to completion is the same for any connected graph on n vertices and is independent of graph structure or the number of edges. It depends only on the initial number of vertices of each colour ($R(0)$, $B(0)$). Whenever a discordant edge is chosen, the number of blue vertices in the graph increases (resp. decreases) by one with probability $1/2$. This is equivalent to an unbiased random walk on the line $(0, 1, ..., n)$ with absorbing barriers, starting from $R(0) = r$ red vertices. Thus $\mathbf{E}T = r(n - r)$ (see Feller [9, XIV.3]).

▶ Remark. *Oblivious protocol.* Let T be the time to consensus in the two-party asynchronous discordant voting process starting from any initial coloring with an equal number of red and blue vertices $R = r, B = n - r$. For any connected n vertex graph, $\mathbf{E}T(\text{Oblivious}) = r(n - r)$.

In stark contrast to the oblivious protocol, the discordant push and pull protocols can exhibit very different expected times to consensus, and which depend strongly on the underlying graph in question.

▶ **Theorem 1.** *Let T be the time to consensus of the asynchronous discordant voting process starting from any initial coloring with an equal number of red and blue vertices $R = B = n/2$. For the complete graph K_n, and for random graphs $G_{n,p}$, $np \geq \log n\sqrt{n}$, $\mathbf{E}T(Push) = \Theta(n \log n)$, and $\mathbf{E}T(Pull) = \Theta(2^n)$.*

For reasons of brevity we do not reproduce the proof for $G_{n,p}$ here, but will make it available in the full version of this paper. The interesting point is that for the complete graph K_n and random graphs $G_{n,p}$ the different protocols give very different expected completion times, which vary from $\Theta(n \log n)$ for push, to $\Theta(n^2)$ for oblivious to $\Theta(2^n)$ for pull. On the basis of this evidence, our initial view was that there should be a meta-theorem of the "push is faster than oblivious, oblivious is faster than pull" type. Intuitively, this is supported by the following argument. Suppose red (R) is the larger colour class. Choosing a discordant vertex uniformly at random, favors the selection of the larger class. In the push process, red vertices push their opinion more often, which tends to increase the size of R. Conversely, the pull process tends to re-balance the set sizes. If R is larger, it is recoloured more often.

If the graph has limited expansion, the behavior of discordant voting differs considerably from the above examples. For the cycle C_n, all three protocols are similar.

▶ **Theorem 2.** *Let T be the time to consensus of the asynchronous discordant voting process starting from any initial coloring with an equal number of red and blue vertices $R = B = n/2$. For any of the Push, Pull or Oblivious protocols on the cycle C_n, $\mathbf{E}T = \Theta(n^2)$.*

At this point we were left with a difficult choice. Either to produce evidence for a relationship of the form $\mathbf{E}T(\text{Push}) = O(\mathbf{E}T(\text{Pull}))$, or to refute it. Mossel and Roch [16] found slow convergence of the iterated prisoners dialemma problem (IPD) on caterpillar trees. Intuitively push voting is aggressive, whereas pull voting is altruistic, and thus similar to cooperation in IPD. Motivated by this, we found the star graph S_n as a counter example.

▶ **Theorem 3.** *Let T be the time to consensus in the two-party asynchronous discordant voting process starting from any initial coloring with an equal number of red and blue vertices $R = B = n/2$. For the star graph S_n, $\mathbf{E}T(Push) = \Theta(n^2 \log n)$, and $\mathbf{E}T(Pull) = O(n^2)$.*

For stars, experiments show a clear difference in $\mathbf{E}T$ for three protocols. For cycles the difference is smaller and depends on the initial colouring. See Fig. 4 of Section 5.1.

A major problem in analysing discordant voting, is that the effect of recolouring a vertex is not always monotone. For each of the graphs studied, the way to bound $\mathbf{E}T$ differs. The proof of the pull voting result for the cycle C_n in particular, is somewhat delicate, and requires an analysis of the optimum of a linear program based on a potential function.

Asynchronous discordant voting model

We next give a formal definition of the discordant voting process. Given a graph $G = (V, E)$, with $n = |V|$. Each vertex $v \in V$ is labelled with an *opinion* $X(v) \in \{0, 1\}$. We call X a *configuration* of opinions. We can think of the opinions as having colours; e.g. red (0) and blue (1), or black (0) and white (1) (see e.g. Figure 2). An edge $e = uv \in E$ is *discordant* if $X(u) \neq X(v)$. Let $K(X)$ denote the set of discordant edges at time t. A vertex v is discordant if it is incident with any discordant edge, and $D(X)$ will denote the set of discordant vertices in X. We consider three random update rules for opinions X_t at time t.

Push: Choose $v_t \in D(X_t)$, uniformly at random, and a discordant neighbour u_t of v_t uniformly at random. Let $X_{t+1}(u_t) \leftarrow X_t(v_t)$, and $X_{t+1}(w) \leftarrow X_t(w)$ otherwise.

Pull: Choose $v_t \in D(X_t)$, uniformly at random, and a discordant neighbour u_t of v_t uniformly at random. Let $X_{t+1}(v_t) \leftarrow X_t(u_t)$, and $X_{t+1}(w) \leftarrow X_t(w)$ otherwise.

Oblivious: Choose $\{u_t, v_t\} \in K(X_t)$ uniformly at random. With probability $1/2$, $X_{t+1}(v_t) \leftarrow X_t(u_t)$, with probability $1/2$, $X_{t+1}(u_t) \leftarrow X_t(v_t)$, and $X_{t+1}(w) \leftarrow X_t(w)$ otherwise.

These three processes are Markov chains on the configurations in G, in which the opinion of exactly one vertex is changed at each step. Assuming G is connected, there are two absorbing states, when $X(v) = 0$ for all $v \in V$, or $X(v) = 1$ for all $v \in V$, where no discordant vertices exist. When the process reaches either of these states, we say that is has converged. Let T be the step at which convergence occurs. Our object of study is $\mathbf{E}T$.

Structure of the paper. In Section 2 we prove results for a Birth-and-Death chain which we call the Push chain. This chain can be coupled with many aspects of the discordant voting process. We then prove Theorems 1, 2 and 3 in that order.

2 Birth-and-Death chains

A Markov chain $(X_t)_{t\geq 0}$ is said to be a Birth-and-Death chain on state space $S = \{0, \ldots, N\}$ if given $X_t = i$ then the possible values of X_{t+1} are $i + 1, i$ or $i - 1$ with probability p_i and q_i respectively. We assume that $q_0 = p_N = 0$, and $p_0 = 1$, $q_N = 1$, and $p_i > 0$, $q_i > 0$ otherwise. Denote by $\mathbf{E}_i T_j$ the expected hitting time of state j starting from state i, i.e. $T_j = \min\{t \geq 0 : X_t = j,\ X_0 = i\}$. We summarize the results we require on Birth-and-Death chains (see Peres, Levin and Wilmer [15, 2.5]).

A probability distribution π satisfies the detailed balance condition, if

$$\pi(i)P(i, j) = \pi(j)P(j, i), \text{ for all } i, j \in S. \tag{1}$$

Birth-and-Death chains with $p_i = P(i, i+1), q_i = P(i, i-1)$ can be shown to satisfy the detailed balance equations. It follows from this, (see e.g. [15]) that

$$\mathbf{E}_{i-1}T_i = \frac{1}{q_i\pi(i)}\sum_{k=0}^{i-1}\pi(k) \tag{2}$$

An equivalent formulation (see [15]) is $\mathbf{E}_0T_1 = 1/p_0 = 1$ and in general

$$E_{i-1}T_i = \sum_{k=0}^{i-1}\frac{1}{p_k}\frac{q_{k+1}\cdots q_{i-1}}{p_{k+1}\cdots p_{i-1}}, \qquad \text{for } i \in \{1,\ldots,N\}. \tag{3}$$

In writing this expression we follow the convention that if $k = i-1$ then $\frac{q_{k+1}\cdots q_{i-1}}{p_{k+1}\cdots p_{i-1}} = 1$ so that the last term is $1/p_{i-1}$. Note also that the final index k on p_k is $k = N-1$, i.e. we never divide by $p_N = 0$.

Starting from state 0, let T_M be the number of transitions needed to reach state M for the first time. For any $M \le N$, we have that $\mathbf{E}_0T_M = \sum_{i=1}^{M}\mathbf{E}_{i-1}T_i$. For example, $\mathbf{E}_0T_1 = \frac{1}{p_0} = 1$ and $\mathbf{E}_0T_2 = 1 + \frac{1}{p_1} + \frac{q_1}{p_0p_1}$ etc. Thus, for $M \ge 1$

$$\mathbf{E}_0T_M = \sum_{i=1}^{M}\mathbf{E}_{i-1}T_i = \sum_{i=1}^{M}\sum_{k=0}^{i-1}\frac{1}{p_k}\prod_{j=k+1}^{i-1}\frac{q_j}{p_j}. \tag{4}$$

We next define a Birth-and-Death chain, the push chain, which features in our analysis. The chain has states $\{0, 1, ..., i, ..., N\}$ where $N = n/2$ (assume $n \ge 2$ even). The transition probabilities from state i given by $P(i, i+1)$, $Q(i, i+1) = 1 - P(i, i+1)$.

Let Z_t be the state occupied by the push chain at step $t \ge 0$. Let $\delta \in \{-1, 0, +1\}$ be fixed. When applying results for the push chain in our proofs, we will state the value of δ we use. The transition probability $p_i = P(i, i+1)$ from $Z_t = i$, is given by

$$p_i = \begin{cases} 1, & \text{if } i = 0 \\ 1/2 + i/n + \delta/n, & \text{if } i \in \{1,\ldots,n/2-1\} \\ 0, & \text{if } i = n/2 \end{cases}. \tag{5}$$

For a proof of the following lemma see [5].

▶ **Lemma 4.** *Let $\mathbf{E}_0T_M$ be the expected hitting time of M in the push chain Z_t starting from state 0. For any $M \le N$,*

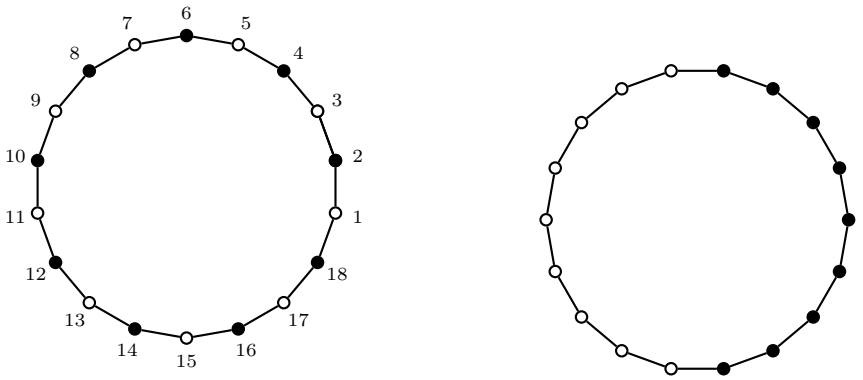
$$\mathbf{E}_0T_M \le 2N\log M + O(1). \tag{6}$$

For any $\sqrt{N} \le M = o(N^{3/4})$, there exists a constant $C > 0$ such that

$$\mathbf{E}_0T_M \ge C(N\log(M/\sqrt{N}) + \sqrt{N}). \tag{7}$$

3 Voting on the complete graph K_n

For the complete graph K_n, the probability B increases at a given step is $B(t)/n$, whereas in the pull process it is $R(t)/n = 1 - B(t)/n$. The chain defined by $Y_t = \max\{R(t), B(t)\} - n/2$ is a Birth-and-Death chain. We study the time that takes Y_t to reach $N = n/2$ starting from 0.



(a) All $X(i) = i \bmod 2$ **(b)** Lower bound configuration

Figure 1 A cycle with $n = 18$. Example colourings.

Push process. For the push model, the process Y_t is identical to the push chain Z_t with transitions p_i given by (5), with $\delta = 0$. The result of Theorem 1 that $\mathbf{E}T(\text{Push}) = \Theta(n \log n)$ follows from Lemma 4.

Pull process. As pull is the opposite of push, the pull process Y_t has transitions given by $\overline{p}_i = 1 - p_i$, i.e. . Thus $\overline{p}_0 = 1$, $\overline{p}_i = 1/2 - i/n$ if $i \in \{1, \dots, N-1\}$, and $\overline{p}_N = 0$.

Let $w_k = \binom{n}{N+k}, k = 0, 1, \dots, N$. Then w_k satisfies the detailed balance equation (1). Hence we have $\pi(k) = w_k/W$, where $W = w_0 + w_1 + \cdots + w_N$. It follows from (2) that

$$\mathbf{E}_{i-1}T_i = \frac{2n}{n+2i} \cdot \frac{1}{\binom{n}{N+i}} \cdot \sum_{k=0}^{i-1} \binom{n}{N+k}.$$

Putting $i = N$ we have

$$\mathbf{E}_{N-1}T_N = \sum_{k=0}^{N-1} \binom{n}{N+k} = \frac{1}{2}\left(2^n - 2 + \binom{n}{N}\right) = \Omega(2^n).$$

On the other hand, an upper bound

$$\sum_{i=1}^{N} \mathbf{E}_{i-1}T_i \le 2 \cdot 2^n \cdot \sum_{i=1}^{N} \frac{1}{\binom{n}{N+i}} = O(2^n),$$

follows from a result of Sury [19], that

$$\sum_{i=1}^{N} \frac{1}{\binom{n}{N+k}} = \frac{n+1}{2^n} \sum_{i=0}^{n} \frac{2^i}{i+1} = O(1).$$

4 Voting on the cycle

An n-cycle G, with $V = [n]$, has $E = \{(i, i+1) : i \in [n]\}$, where we identify vertex $n+i$ with vertex i. See Fig. 1(i).

If $X(i) \neq X(i+1) = X(i+2) = \cdots = X(j) \neq X(j+1)$, we say $i+1, i+2, \dots, j$ is a *run* of vertices of length $(j-i)$ $(1 \le j - i < n)$. Note that the number of runs is equal to the number of discordant edges $k(X)$. Also k is even, since red and blue runs must alternate,

so we can write $r(X) = \frac{1}{2}k(X)$, and $k_0 = 2r_0 = k(X_0)$. Thus $r(X)$ is the number of paths of a given colour. A *singleton* is a run of length 1. Since they lie in two discordant edges, singletons require special treatment. Let $s(X)$ denote the number of singletons. There are $\kappa = 2k - s$ discordant vertices, so $k \le \kappa \le 2k$.

We wish to determine the convergence time T for an arbitrary configuration X_0 of the push or pull process to reach an absorbing state X_T with $X_T(i) = X_T(1)$ $(i \in [n])$. In this process, the run lengths behave rather like symmetric random walks on the line. However, an analysis using classical random walk techniques [9] seems problematic. There are two main difficulties. Firstly, the k "walks" (run lengths) are correlated. If a run is long, the adjacent runs are likely to be shorter, and vice versa. Secondly, when the recoloured vertex is a singleton, the three adjacent runs are combined, so three walks suddenly merge into one. One of the three runs is a singleton, but the other two may have arbitrary lengths. We use the random walk view only for a lower bound on the convergence time.

▶ **Lemma 5.** *Let G is an n-cycle, with $n = 2N$ even, and suppose the process starts with $X_0(i) = 0$ $(i = 1, \ldots, N)$, $X_0(i) = 1$ $(i = N+1, \ldots, n)$, then $\mathbf{E}[T] = \Omega(n^2)$.*

Let L_t be the length of (say) the red run at step t, so $L_0 = N$, (see Fig. 1(ii)), and $L_T \in \{0, n\}$. The number of runs $k(X_t)$ can only be reduced from two to zero if either $L_t = 1$ or $L_t = n-1$, when one of the runs is a singleton. Up to this point, L_t is a symmetric simple random walk and the push and pull processes proceed identically. Thus $\mathbf{E}[T]$ is bounded below by the expected time for a symmetric simple random walk started at N to reach either 1 or $(n-1)$. By Remark 1, $\mathbf{E}[T] \ge (N-1)^2 = \Omega(n^2)$.

4.1 Upper bound for push voting: Proof that $\mathbf{E}[T] = O(n^2)$

Let the k runs in X have lengths $\ell_1, \ell_2, \ldots, \ell_k$ respectively, thus $\sum_{i=1}^{k} \ell_i = n$. Thus T is the first t for which $k(X_t) = r(X_t) = 0$, (a cycle is not a path). For an upper bound on $\mathbf{E}[T]$, we define a *potential function*

$$\psi(X) \;=\; \sum_{i=1}^{k} \sqrt{\ell_i}\,,$$

where $\psi(X) = 0$ if and only if $k(X) = 0$. The important feature of ψ is that it is a separable and strictly concave function of the ℓ_i $(i \in [k])$.

▶ **Lemma 6.** *For any configuration X on the n-cycle with k runs, $\psi(X) \le \sqrt{kn}$.*

Proof. If $k = 0$, this is clearly true. Otherwise, if $k \ge 2$, by concavity we have $\psi(X)/k = \frac{1}{k}\sum_{i=1}^{k}\sqrt{\ell_i} \le \sqrt{\frac{1}{k}\sum_{i=1}^{k}\ell_i} = \sqrt{n/k}$, so $\psi(X) \le \sqrt{kn}$. ◀

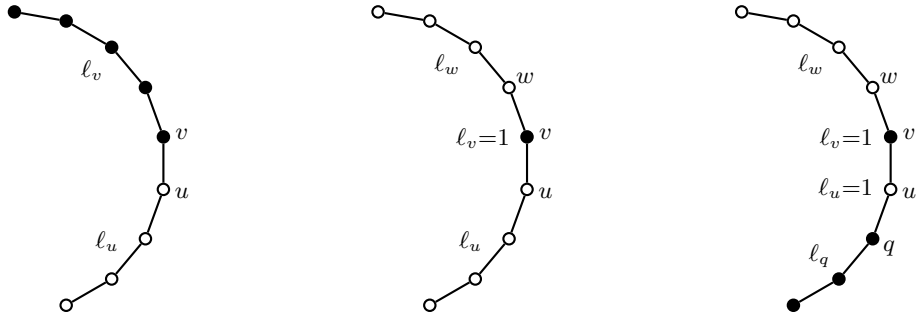
Observe that $k(X_{t+1}) = k(X_t)$ at step t of either the push or pull process, unless the recoloured vertex is a singleton, in which case we may have $k(X_{t+1}) = k(X_t) - 2$. Thus $\{t : k(X_t) = 2r\}$ is an interval $[t_r, t_{r-1})$, which we will call *phase* r of the process.

Let $v_t = v \in D(X_t)$ be the active vertex, i.e. the vertex selected to push in the push rule, or pull in the pull rule. Let δ_v be the expected change in ψ, i.e.

$$\delta_v \;=\; \mathbf{E}[\,\psi(X_{t+1}) - \psi(X_t) \mid v_t = v\,].$$

If there are $\kappa = 2k - s$ discordant vertices, the total expected change δ in ψ is

$$\delta \;=\; \mathbf{E}[\psi(X_{t+1}) - \psi(X_t)] \;=\; \frac{1}{\kappa}\sum_{v \in D} \delta_v. \tag{8}$$



(A) u and v not singleton **(B)** u not singleton, v singleton **(C)** u and v both singleton

Figure 2 Cases for discordant edge uv.

We will show that δ is negative, so $\psi(X_t)$ is monotonically decreasing with t, in expectation. Unfortunately we cannot simply bound δ_v for each $v \in D$, since it is possible to have $\delta_v > 0$. Thus we will consider discordant *edges*. We partition the set K of discordant edges uv into three subsets, Note that k can change only if $uv \in B \cup C$.

(A) $A = \{uv : u \text{ and } v \text{ not singleton}\}$;
(B) $B = \{uv : u \text{ not singleton, } v \text{ singleton}\}$;
(C) $C = \{uv : u \text{ and } v \text{ both singleton}\}$.

See Fig. 2. Let ℓ_z be the length of the run containing discordant vertex z, for $z \in \{u, v, w, q\}$.

Now let

$$\lambda_{uv} = \begin{cases} \sqrt{\ell_u} + \sqrt{\ell_v}, & uv \in A\,; \\ \sqrt{\ell_u} + \frac{1}{2}\sqrt{\ell_v}, & uv \in B\,; \\ \frac{1}{2}\sqrt{\ell_u} + \frac{1}{2}\sqrt{\ell_v}, & uv \in C\,. \end{cases} \qquad \delta_{uv} = \begin{cases} \delta_u + \delta_v, & uv \in A\,; \\ \delta_u + \frac{1}{2}\delta_v, & uv \in B\,; \\ \frac{1}{2}\delta_u + \frac{1}{2}\delta_v, & uv \in C\,. \end{cases}$$

Each singleton is in two discordant edges, all other discordant vertices in one, and each run is bounded by two discordant vertices. Therefore

$$\psi = \tfrac{1}{2}\sum_{v\in D}\sqrt{\ell_v} = \sum_{uv\in K}\lambda_{uv}\,, \qquad \delta = \frac{1}{\kappa}\sum_{v\in D}\delta_v = \frac{1}{\kappa}\sum_{uv\in K}\delta_{uv}\,.$$

We will show that $\delta_{uv} < 0$ for all $uv \in K$. For the proof of the following lemma see [5].

▶ **Lemma 7.** *For all three cases (A)–(C), and for all $uv \in K$,*
For push voting, $\delta_{uv} < -\frac{1}{5}(\ell_v^{-3/2} + \ell_u^{-3/2})$. For pull voting, $\delta_{uv} < -\frac{1}{10}(\ell_v^{-3/2} + \ell_u^{-3/2})$.

The following proof that $\mathbf{E}[T] = O(n^2)$ is for push voting. The upper bound on $\mathbf{E}[T]$ for pull voting is at most twice that for push. Using Lemma 7 we evaluate δ in (8).

$$\delta = \frac{1}{\kappa}\sum_{v\in D}\delta_v = \frac{1}{\kappa}\sum_{uv\in K}\delta_{uv} \le -\frac{1}{5\kappa}\sum_{uv\in K}(\ell_v^{-3/2} + \ell_u^{-3/2}) < -\frac{1}{5\kappa}\sum_{v\in D}\ell_v^{-3/2}\,.$$

Thus

$$\mathbf{E}[\psi(X_{t+1})] < \psi(X_t) - \frac{1}{5\kappa}\sum_{v\in D}\ell_v^{-3/2}\,.$$

Since $f(x) = x^{-3}$ is a convex function, $\mathbf{E}[f(X)] \ge f(\mathbf{E}[X])$ by Jensen's inequality [20, 6.6], so

$$\frac{1}{\kappa}\sum_{v\in D}\ell_v^{-3/2} \ge \Big(\frac{1}{\kappa}\sum_{v\in D}\sqrt{\ell_v}\Big)^{-3} = \Big(\frac{\kappa}{2\psi(X_t)}\Big)^3 \ge \Big(\frac{k}{2\psi(X_t)}\Big)^3,$$

Therefore,

$$\mathbf{E}[\psi(X_{t+1})] \;<\; \psi(X_t) - \frac{1}{5}\Big(\frac{k}{2\psi(X_t)}\Big)^3 \;=\; \psi(X_t) - \frac{k^3}{40\psi(X_t)^3}\,. \tag{9}$$

Recall that for $r \in [r_0]$, phase r of the process, during which the number of runs is $k = 2r$, is the interval $[t_r, t_{r-1})$. During phase r, by Lemma 6, $\psi(X_t) \le \sqrt{kn}$. Using this in (9) gives

$$\mathbf{E}[\psi(X_{t+1})] - \psi(X_t) \;\le\; -\tfrac{1}{40}k^3/(kn)^{3/2} \;=\; -\tfrac{1}{40}(k/n)^{3/2}\,. \tag{10}$$

Let $\gamma_r = \frac{1}{40}(2r/n)^{3/2}$. Then (10) implies that $Y_t = \psi(X_t) + (t - t_r)\gamma_r$ is a supermartingale [20, 10.3] during phase r, and t_{r-1} is a stopping time. Let $\varphi_r = \mathbf{E}[\psi(X_{t_r})]$, and let $m_r = \mathbf{E}[t_{r-1} - t_r]$. The optional stopping theorem [20, 10.10] implies that

$$\varphi_{r-1} + \gamma_r m_r \;=\; \mathbf{E}[\psi(X_{t_{r-1}}) + \gamma_r(t_{r-1} - t_r)] \;\le\; \mathbf{E}[\psi(X_{t_r})] \;=\; \varphi_r\,,$$

which implies

$$\varphi_r - \varphi_{r-1} \;\ge\; \gamma_r m_r \;=\; \tfrac{1}{40}m_r(2r/n)^{3/2} \qquad (r \in [r_0])\,. \tag{11}$$

Note, in particular, that $\varphi_r \ge \varphi_{r-1}$ for all $r \in [r_0]$. When $r_0 = \frac{1}{2}k(X_0)$, $t_{r_0} = 0$ and, since $r(X_T) = k(X_T) = 0$, when $t_0 = T$ then $\varphi_0 = 0$.

Let $x_r = \varphi_r - \varphi_{r-1} \ge 0$, for $r \in [r_0]$, so $\varphi_r = \sum_{i=1}^{r} x_j \le \sqrt{2rn}$. Also, from (11), we have $m_r \le 40x_r(n/2r)^{3/2} = 10\sqrt{2}\,n^{3/2}x_r/r^{3/2}$, so $\mathbf{E}[T] = \sum_{j=1}^{r_0} m_j < 10\sqrt{2}\,n^{3/2}\sum_{j=1}^{r_0} x_r/r^{3/2}$.

Thus $E[T]$ is bounded above by $T^\star$, the optimal value of the following linear program.

$$\begin{aligned} T^\star \;&=\; \max\; 10\sqrt{2}\,n^{3/2}\textstyle\sum_{r=1}^{r_0} x_r/r^{3/2} \\ \text{such that}\;\; \textstyle\sum_{j=1}^{r} x_j \;&\le\; \sqrt{2rn} && (r \in [r_0]) \\ x_j \;&\ge\; 0 && (j \in [r_0])\,. \end{aligned} \tag{12}$$

This linear program can be solved by a greedy procedure.

▶ **Lemma 8.** *Let $0 < b_1 < b_2 < \cdots < b_\nu$ and $c_1 > c_2 > \cdots > c_\nu > 0$. Then the linear program $\max \sum_{j=1}^{\nu} c_j x_j$ subject to $\sum_{j=1}^{r} x_j \le b_r$, $x_r \ge 0$ $(r \in [\nu])$ has optimal solution $x_1 = b_1$, $x_j = b_j - b_{j-1}$ $(j = 2, 3, \ldots, \nu)$.*

Proof. This solution has objective function value $c_1b_1 + c_2(b_2 - b_1) + \cdots + c_\nu(b_\nu - b_{\nu-1})$. The dual linear program is $\min \sum_{i=1}^{\nu} b_i y_i$ subject to $\sum_{i=j}^{\nu} y_i \ge c_j$, $y_j \ge 0$ $(j \in [\nu])$, and has feasible solution $y_\nu = c_\nu$, $y_j = c_j - c_{j+1}$ $(j \in [\nu - 1])$. Then the dual objective function has value $b_\nu c_\nu + b_{\nu-1}(c_{\nu-1} - c_\nu) + \cdots + b_1(c_1 - c_2)$. However,

$$c_1b_1 + c_2(b_2 - b_1) + \cdots + c_\nu(b_\nu - b_{\nu-1}) \;=\; b_\nu c_\nu + b_{\nu-1}(c_{\nu-1} - c_\nu) + \cdots + b_1(c_1 - c_2)\,.$$

Since the objective function values are equal, it follows that the two solutions are optimal in the primal and dual respectively. ◀

Thus, the optimal solution to (12) is $x_r = \sqrt{2nr} - \sqrt{2n(r-1)} = \sqrt{2nr}(1 - \sqrt{1 - 1/r}) \le \sqrt{2n/r}$, for $r \in [r_0]$, since $1 - y \le \sqrt{1-y}$ for $0 \le y \le 1$. Thus

$$T^\star \;\le\; 10\sqrt{2}\,n^{3/2}\sum_{j=1}^{r_0} x_r/r^{3/2} \;\le\; 10\sqrt{2}\,n^{3/2}\sum_{j=1}^{r_0}\sqrt{2n}/\big(\sqrt{r}\,r^{3/2}\big) \;=\; 20n^2\sum_{r=1}^{r_0} 1/r^2 \;<\; (10\pi^2/3)n^2\,,$$

since $\sum_{r=1}^{\infty} 1/r^2 = \pi^2/6$. Thus we have an absolute bound of $\mathbf{E}[T] = O(n^2)$.

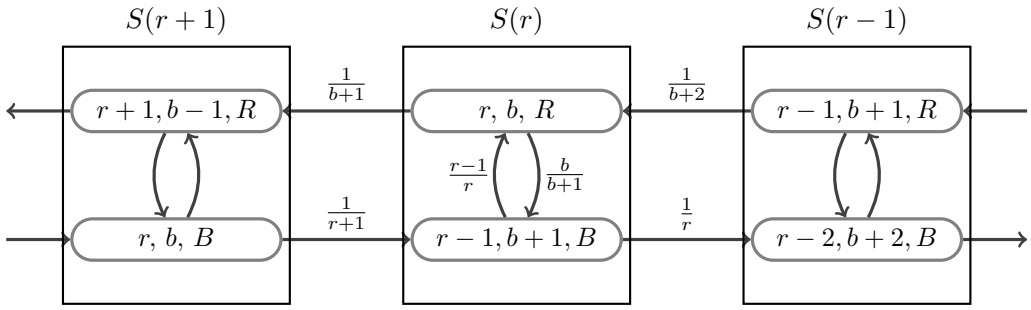


Figure 3 Star graph: Pseudo-states for the push process.

5 Theorem 3: Voting on the star graph S_n

In this section we prove $\mathbf{E}T(\text{Push}) = \Theta(n^2 \log n)$. The result that $\mathbf{E}T(\text{Pull}) = O(n^2)$ is given in [5].

Let (r, b, X) denote the coloring of the star graph S_n on n vertices in which there are r red vertices $b = n - r$ blue vertices. The central vertex has colour $X \in \{R, B\}$. In the case of the push process, the transitions from state (r, b, R) are to state $(r+1, b-1, R)$ with probability $1/(b+1)$ and to state $(r-1, b+1, B)$ with probability $b/(b+1)$. The transitions from state $(r-1, b+1, B)$ are to (r, b, R) with probability $(r-1)/r$ and to $(r-2, b+2, B)$ with probability $1/r$. For the purposes of discussion we group the states $(r, R) = (r, b, R)$ and $(r-1, B) = (r-1, b+1, B)$ into a single pseudo-state $S(r)$.

The transitions probabilities within or between $S(r+1)$ or $S(r-1)$ are shown in Figure 3, and are derived as follows. Let $X, Y \in \{R, B\}$. For a particle occupying a state (of colour) X in $S(r)$ let $P_X(Y, r)$ be the probability of exit from $S(r)$ via state Y. For example $P_R(R, r)$ is the probability that a particle starting at (r, R) eventually exits from $S(r)$ via state (r, R) to state $(r+1, R)$ in $S(r+1)$. Thus

$$P_R(R, r) = \frac{1}{b+1}\left(1 + \frac{b}{b+1}\frac{r-1}{r} + \cdots + \left(\frac{b}{b+1}\frac{r-1}{r}\right)^k + \cdots\right),$$

so that

$$P_R(R, r) = \frac{1}{b+1}\frac{1}{1 - [b(r-1)/(b+1)r]} = \frac{r}{n}.$$

Similarly let $P_B(R, r)$ be the probability that a particle currently at $(r-1, B)$ in $S(r)$ moves from $S(r)$ to $(r+1, R)$ in $S(r+1)$. Then

$$P_B(R, r) = \frac{r-1}{r}P_R(R, r) = \frac{r-1}{n}.$$

In summary, starting from state $X \in \{R, B\}$ of $S(r)$, for $1 \le r \le n-1$ the transition probability $p_X(r)$ from $S(r)$ to $S(r+1)$ (resp. transition probability $p_X(b)$ from $S(r)$ to $S(r-1)$) is given by

$$p_X(r) = \frac{r - 1_{(X=B)}}{n}, \qquad p_X(b) = \frac{b + 1_{(X=B)}}{n}. \tag{13}$$

States $(0, B)$ (i.e. $S(0)$) and (n, R) (i.e. $S(n)$) are absorbing.

Let $i = \max(r, b) - n/2$. To obtain lower and upper bounds on the number of transitions *between pseudo-states* $S(r)$ *before absorption*, we can couple the process with a biassed

random walk on the line $L = \{0, 1, ..., n/2\}$ with a reflecting barrier at 0 and an absorbing barrier at $n/2$. We assume n is even here. For $0 < i < n/2$, let p_i be the probability of a transition from i to $i+1$ on L, and let $q_i = 1 - p_i$ be the probability of a transition from i to $i-1$. It follows from (13) that to obtain bounds on the number of transitions between pseudo-states $S(r)$ before absorption we can use a value of p_i given by

$$p_i = 1/2 + (i+1)/n \quad \text{Lower bound}, \qquad p_i = 1/2 + (i-1)/n \quad \text{Upper bound}. \tag{14}$$

We next consider the number of loops, for example $(r, R) \to (r-1, B) \to (r, R)$, made within $S(r)$ before exit. For a particle starting from state X of $S(r)$ let $C_{XY} = C_{XY}(r)$ be the number of loops before exit at state Y. Let $\lambda = \frac{b}{b+1}\frac{r-1}{r}$ and $\rho = \lambda/(1-\lambda)^2$, then

$$\mathbf{E}C_{RR} = \sum_{k\geq 0} \frac{1}{b+1} k\lambda^k = \frac{1}{b+1}\frac{\lambda}{(1-\lambda)^2} = \rho\frac{1}{b+1}.$$

Similarly,

$$\mathbf{E}C_{BR} = \rho\frac{r-1}{r(b+1)}, \qquad \mathbf{E}C_{RB} = \rho\frac{b}{r(b+1)}, \qquad \mathbf{E}C_{BB} = \rho\frac{1}{r}.$$

The conditional expectations $\mu_{XY}(r) = \mathbf{E}C_{XY}(r)/P_X(Y, r)$ are given by

$$\mu_{XY}(r) = \begin{cases} \rho\frac{n}{r}\frac{1}{b+1}, & XY = RR \\ \rho\frac{n}{r}\frac{1}{b+1}, & XY = BR \\ \rho\frac{n}{n-r}\frac{b}{r(b+1)}, & XY = RB \\ \rho\frac{n}{n-r+1}\frac{1}{r}, & XY = BB \end{cases}. \tag{15}$$

The value of $\rho = (rb(r-1)(b+1))/n^2$. In particular if $b, r = (1 + o(1))n/2$ then, whatever colours X, Y

$$\mu_{XY}(r) = (1 + o(1))\frac{n}{4}. \tag{16}$$

Let $N = n/2$. Starting from $r = b = N$ let T'_N be the number of transitions between states $S(r)$ to reach $\max(r, b) = N + n/2$. Referring to (14), we consider a biassed random walk with transition probabilities of $Z = \max\{r, b\} - n/2$ given by

$$p_i = \begin{cases} 1, & \text{if } i = 0 \\ 1/2 + i/n + \delta/n, & \text{if } i \in \{1, \ldots, n/2 - 1\} \\ 0, & \text{if } i = n/2 \end{cases}, \tag{17}$$

where we set $\delta = 1$ for a lower bound on the number of steps T' to absorption, and $\delta = -1$ for an upper bound. The walk in (17) is the push chain Z_t with transitions given by (5) as analysed Section 2. Referring to (5) and (4) we set $\delta = 0$ for a lower bound on $\mathbf{E}_0 T_M$. For $M = N^{3/4}$, from Lemma 4,

$$\mathbf{E}_0 T_M \geq \Theta(1) \sum_{i=\sqrt{N}}^{M} \frac{N}{i} \geq \Theta(N) \log \frac{M}{\sqrt{N}} = \Theta(n \log n).$$

For all states $i = \sqrt{N}, ..., N^{3/4}$, the corresponding value of $r = (1 + o(1))n/2$. Referring to (16), whatever the type of transition XY between $S(r)$ and neighbouring states, $\mu_{XY}(r) = (1 + o(1))n/4$. Let $\mu = \min_{X,Y}(\mu_{XY}(r) : n/2 \leq r \leq M)$, then $\mu \geq n/5$. As $\mathbf{E}_0 T_N \geq \mathbf{E}_0 T_M = \Theta(n \log n)$ we have that

$$\mathbf{E}T(Push) \geq \mu\, \mathbf{E}_0 T_M = \Omega(n^2 \log n).$$

The upper bound follows by a similar argument. Put $\delta = -1$ in (5), and use Lemma 4.

5.1 Discordant voting: Simulation results for star graph and cycle

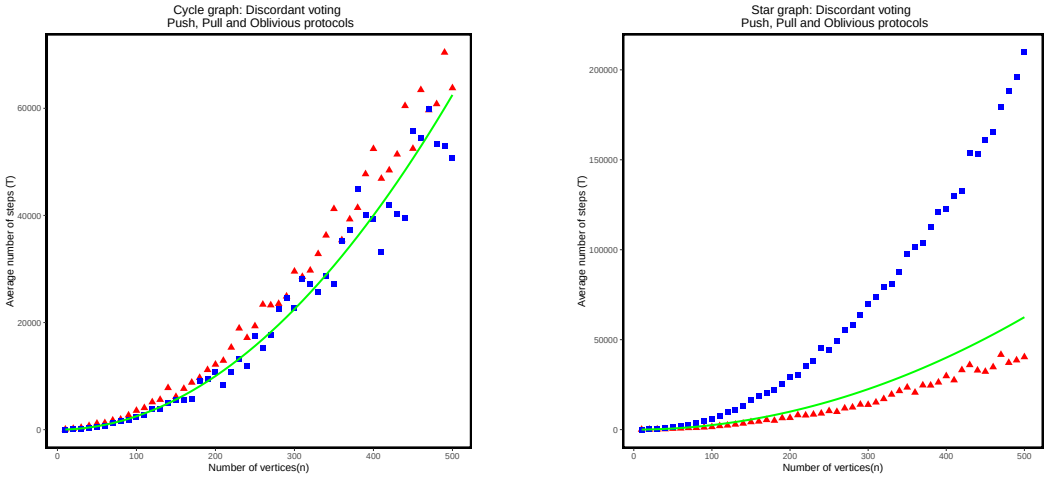


Figure 4 Legend: Push (Square), Pull (Triangle), Oblivious $\mathbf{E}T = n^2/4$ (Solid line). Left plot: Cycle, initial colouring alternating red-blue (see Fig.1(i)), Right plot: Star graph, random colouring $R(0) = B(0) = n/2$. Each plot point consists of at least 15 replications.

References

1 D. Aldous and J. Fill. Reversible markov chains and random walks on graphs, 2002. Unfinished monograph, recompiled 2014, available at `http://www.stat.berkeley.edu/~aldous/RWG/book.html`.

2 J. Aspnes and E. Ruppert. An introduction to population protocols. In *Middleware for Network Eccentric and Mobile Applications*, pages 97–120. Springer, 2009.

3 R. Basu and A. Sly. Evolving voter model on dense random graphs. *arXiv preprint. arXiv:1501.03134*, 2015.

4 S. Brahma, S. Macharla, S. Pal, and S. Singh. Fair leader election by randomized voting. In *Distributed Computing and Internet Technology*, pages 22–31. Springer, 2004.

5 C. Cooper, M. Dyer, A. Frieze, and N. Rivera. Discordant voting processes on finite graphs (full version). *arXiv preprint. arxiv.org/abs/1604.06884*, 2015.

6 C. Cooper, R. Elsasser, H. Ono, and T. Radzik. Coalescing random walks and voting on connected graphs. *SIAM Journal on Discrete Mathematics*, 27(4):1748–1758, 2013.

7 C. Cooper, A. Frieze, and T. Radzik. Multiple random walks in random regular graphs. *SIAM Journal on Discrete Mathematics*, 23(4):1738–1761, 2009.

8 R. Durrett, J. Gleeson, A. Lloyd, P. Mucha, F. Shi, D. Sivakoff, J. Socolar, and C. Varghese. Graph fission in an evolving voter model. *Proceedings of the National Academy of Sciences*, 109(10):3682–3687, 2012.

9 W. Feller. *An introduction to probability theory and its applications*, volume 2. John Wiley & Sons, 2008.

10 D. Gifford. Weighted voting for replicated data. In *Proceedings of the seventh ACM symposium on Operating systems principles*, pages 150–162. ACM, 1979.

11 T. Gross and H. Sayama. *Adaptive networks*. Springer, 2009.

12 Y. Hassin and D. Peleg. Distributed probabilistic polling and applications to proportionate agreement. *Information and Computation*, 171(2):248–268, 2001.

13 P. Holme and M. Newman. Nonequilibrium phase transition in the coevolution of networks and opinions. *Physical Review E*, 74(5):056108, 2006.

14 B. Johnson. *Design & analysis of fault tolerant digital systems*. Addison-Wesley Longman Publishing Co., Inc., 1988.

15 D. Levin, Y. Peres, and E. Wilmer. *Markov chains and mixing times*. AMS Bookstore, 2009.

16 E. Mossel and S. Roch. Slow emergence of cooperation for win-stay lose-shift on trees. *Machine Learning*, 67(1-2):7–22, 2007.

17 T. Nakata, H. Imahayashi, and M. Yamashita. Probabilistic local majority voting for the agreement problem on finite graphs. In *Computing and Combinatorics*, pages 330–338. Springer, 1999.

18 H. Sayama, I. Pestov, J. Schmidt, B. Bush, C. Wong, J. Yamanoi, and T. Gross. Modeling complex systems with adaptive networks. *Computers & Mathematics with Applications*, 65(10):1645–1664, 2013.

19 B. Sury. Sum of the reciprocals of the binomial coefficients. *European Journal of Combinatorics*, 14(4):351–353, 1993.

20 D. Williams. *Probability with martingales*. Cambridge University Press, 1991.

21 Deng X and C. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19(2):257–266, 1994.